

EC2x&EG9x

Voice Over USB and UAC

Application Note

LTE Module Series

Rev. EC2x&EG9x_Voice_Over_USB_and_UAC_Application_Note_V1.0

Date: 2019-02-18

Status: Released



Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.

7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: info@quectel.com

Or our local office. For more information, please visit:

<http://www.quectel.com/support/sales.htm>

For technical support, or to report documentation errors, please visit:

<http://www.quectel.com/support/technical.htm>

Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL WIRELESS SOLUTIONS CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

Copyright © Quectel Wireless Solutions Co., Ltd. 2019. All rights reserved.

About the Document

History

Revision	Date	Author	Description
1.0	2019-02-18	Zichar ZHANG/ Ketn WANG	Initial

Contents

About the Document	2
Contents	3
Table Index.....	4
Figure Index	5
1 Introduction	6
2 Overview	7
3 Description of AT Command	9
4 Operation Procedures	11
4.1. Operation Procedures of Voice over USB.....	11
4.2. Operation Procedures of UAC	13
4.3. Voice over USB Use in Upper Machine	15
4.4. UAC Use in Upper Machine	15
4.4.1. UAC Use in Windows Devices.....	15
4.4.2. UAC Use in Linux Devices	15
5 Appendix A References.....	18

Table Index

TABLE 1: RELATED DOCUMENTS.....	18
TABLE 2: TERMS AND ABBREVIATIONS.....	18

Figure Index

FIGURE 1: VOICE PATH OF VOICE OVER USB FUNCTION	8
FIGURE 2: VOICE PATH OF UAC FUNCTION	8
FIGURE 3: OPERATION PROCEDURES OF VOICE OVER USB FUNCTION.....	12
FIGURE 4: OPERATION PROCEDURES OF UAC FUNCTION	14

1 Introduction

This document mainly introduces the general operation procedures of Voice over USB and UAC (USB Audio Class) functions of EC2x&EG9x series modules.

This document is applicable to following Quectel modules.

- EC2x (including EC25, EC21, EC20 R2.0 and EC20 R2.1)
- EG9x (including EG91 and EG95)

2 Overview

The voice paths of Voice over USB and UAC functions are shown in following **Figure 1** and **2**, respectively.

The difference between Voice over USB and UAC functions is the data interface in host device. In Voice over USB mode, the device provides a serial port virtualized by the USB interface to transfer the PCM raw data, which needs to be processed by customers. UAC uses another standard class of USB interface: USB audio class. In UAC mode, the module is recognized as a virtual sound card in the host device, and an “AC Interface” device appears in the host device manager. Standard third-party audio library, such as *alsa-lib* can be used to run audio application on Linux device (host device), and some third-party tools can also be used to play/record the application on Windows devices.

In voice call status, the module obtains voice data through air interface. DSP decodes the data to voice PCM stream, then the stream will be transferred to device by USB bus. So audio application in the host device can get PCM data through the virtual serial port or virtual sound card, and transfer voice data to the speaker. Similarly, after the voice is recorded by microphone on the device, data will be transferred in an opposite direction. Please note that Voice over USB and UAC functions of EC2x&EG9x modules only support mono, and the PCM data must be set at sample rate of 8KHz and in 16-bit linear format.

PCM application on the device should control the sending speed of PCM data strictly to ensure the continuity of voice data. During the course of operating UAC function, the device will receive 640 bytes of PCM data from the module every 40ms. PCM application should receive these data and process them as soon as possible; meanwhile, PCM application obtains microphone recording data from audio process unit. It is required that PCM application should send 1600 bytes of voice data to the module at an interval of 100ms.

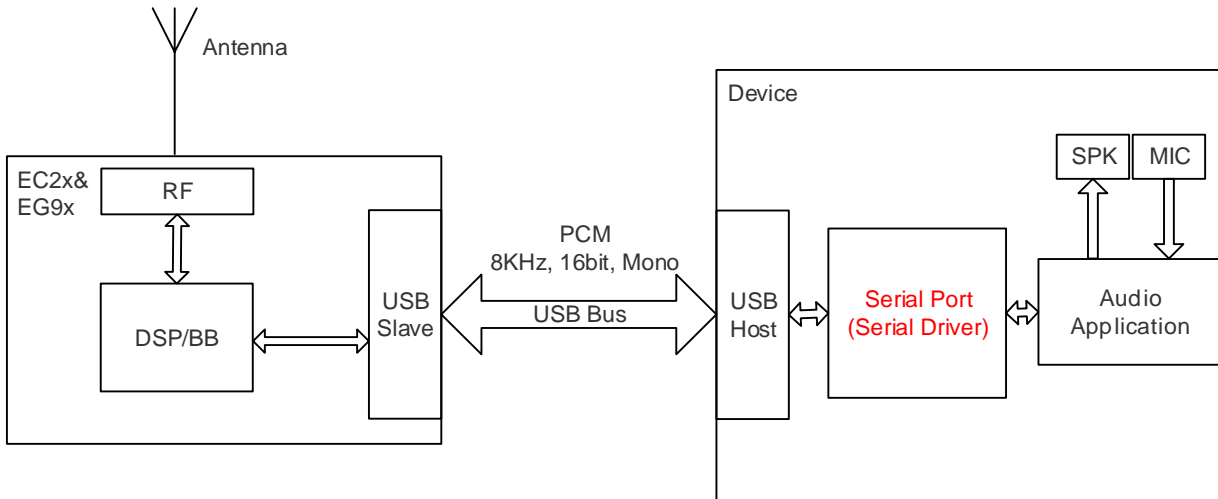


Figure 1: Voice Path of Voice over USB Function

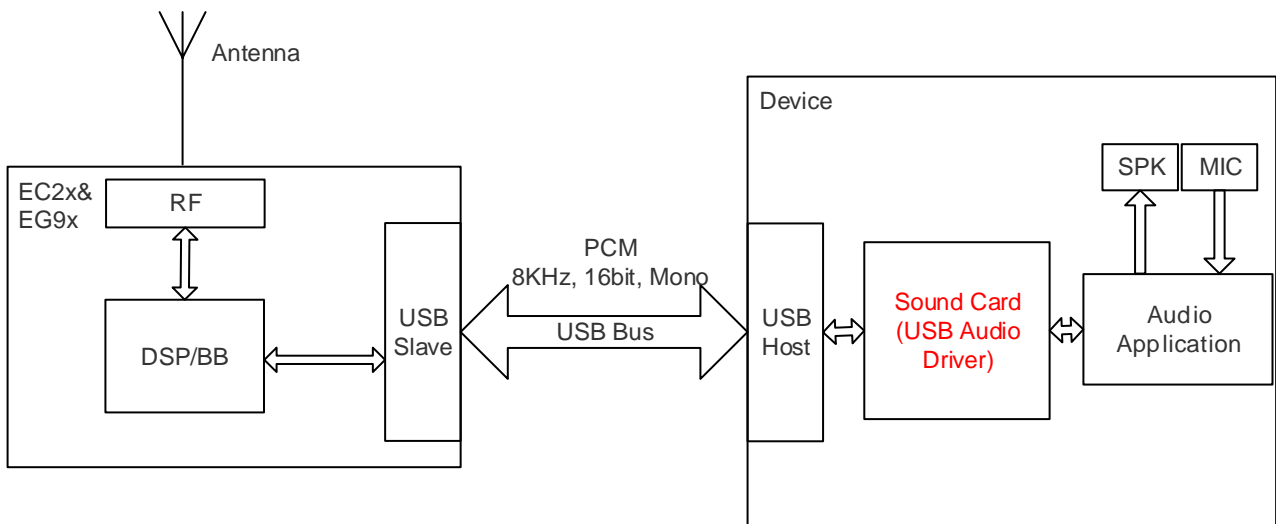


Figure 2: Voice Path of UAC Function

3 Description of AT Command

EC2x&EG9x modules provide **AT+QPCMV** command to enable/disable Voice over USB or UAC function. After the function is enabled, when a call is dialed, the voice data from the opposite side will be decoded to PCM data by module, and then be outputted to device through the USB port which has been set. In the meantime, the device writes the PCM data to the port and the data will be transferred to the other end of the calling device via network. When the call is finished, there is no any data to be outputted, and the inputted data is invalid. The outputted and inputted PCM data must be set at the sample rate of 8KHz and in 16-bit linear format.

AT+QPCMV Enable/Disable Voice over USB or UAC Function	
Test Command AT+QPCMV=?	Response +QPCMV: (0,1),(0,2) OK
Read Command AT+QPCMV?	Response +QPCMV: <enable>[,<option>] OK
Write Command AT+QPCMV=<enable>[,<option>]	Response OK ERROR
URC	+QPCMV: 0 +QPCMV: 1
Reference	

Parameter

<enable>	Enable/disable Voice over USB or UAC function <u>1</u> Enable 0 Disable
<option>	Configure the port or sound card for PCM data transmission <u>0</u> USB NMEA port 1 Debug UART with 230400bps baud rate 2 USB sound card

NOTES

1. Please ensure that the current port is not used for other functions when configuring parameter **<option>**. In addition, when using USB NMEA port, the port should be configured via **AT+QGPSCFG="outport","none"** to prevent GNSS from using the port. Similarly, when using debug UART, it cannot be used as a DM port at the same time.
2. The settings take effect immediately. The parameters are not saved and will restore to default values after the module is restarted.
3. If the modem outputs URC **+QPCMV: 0**, it means the modem is busy, cannot receive more PCM data, and stop sending PCM data to it. If the modem outputs URC **+QPCMV: 1**, it means the modem is ready to receive more PCM data.
4. If UAC mode is used, USB audio device must be enabled via **AT+QCFG="USBCFG"** command.

4 Operation Procedures

This chapter describes the detailed operation procedures of Voice over USB and UAC functions of EC2x&EG9x modules.

4.1. Operation Procedures of Voice over USB

The following steps and figure show the detailed operation procedures of Voice over USB function.

- Step 1:** Power on the module, and it will connect to the network automatically.
- Step 2:** Execute **AT+QPCMV=1,0** on the USB AT port to enable Voice over USB function, and choose USB NMEA port as the input and output port of PCM data.
- Step 3:** ATD dials a call or ATA accepts the incoming call on the USB AT port.
- Step 4:** Audio application on the device will receive 640 bytes of PCM data through USB NMEA port every 40ms, and process (broadcast or save) the received data immediately.
- Step 5:** Audio application on the device will obtain microphone recording data through audio process unit. It is required that the PCM application should send 1600 bytes of voice data to USB NMEA port at an interval of 100ms.
- Step 6:** Execute **ATH** command to end the phone call, and USB NMEA port will stop outputting PCM data.
- Step 7:** Execute **AT+QPCMV=0** to disable Voice over USB function on the USB AT port.

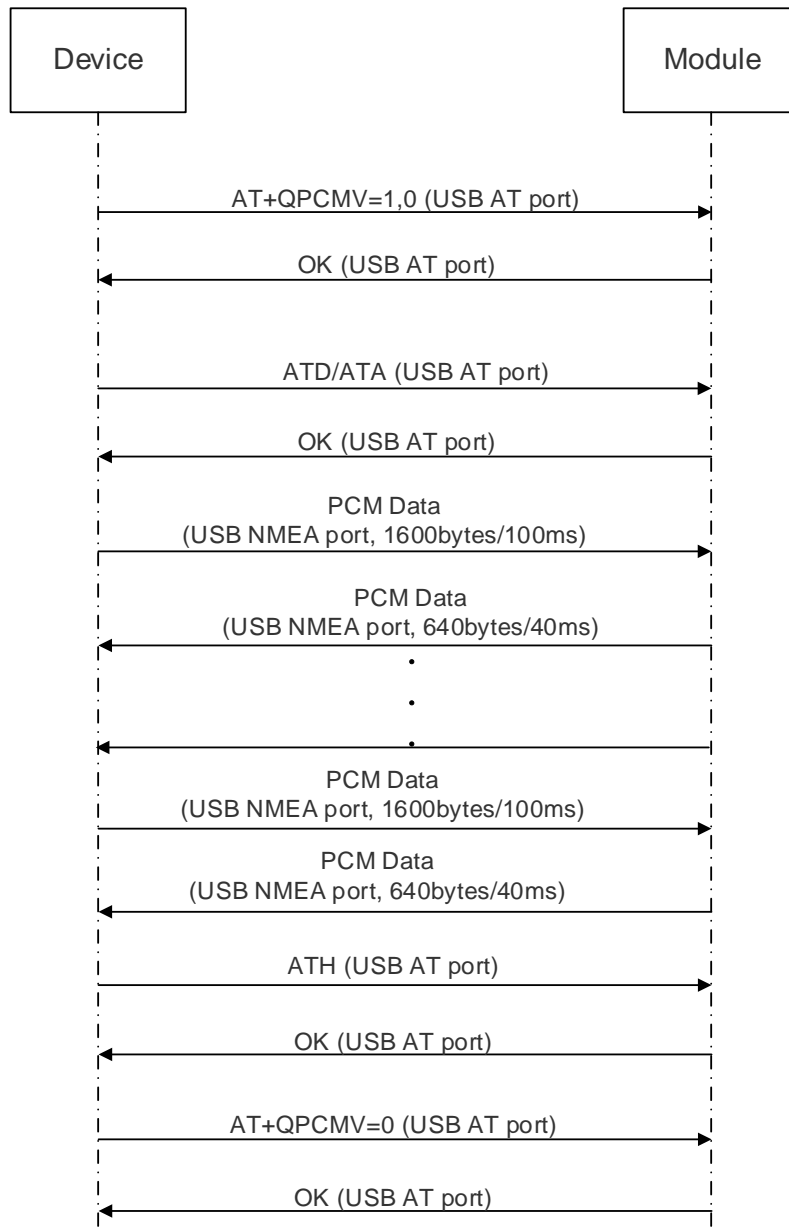


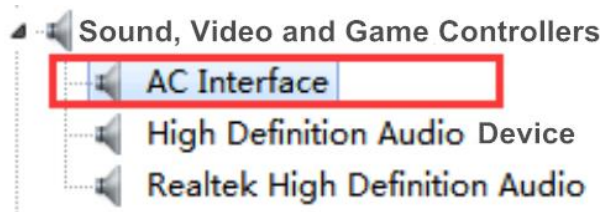
Figure 3: Operation Procedures of Voice over USB Function

4.2. Operation Procedures of UAC

The following steps and figure show the detailed operation procedures of UAC function.

Step 1: Power on the module, and it will connect to the network automatically.

Step 2: Execute **AT+QCFG="USBCFG",0x2C7C,0x0125,x,x,x,x,x,1** on the USB AT port to enable UAC device of module ("x" in the commands means variable value, which can only be 0 or 1). Setting the seventh function parameter in the command to 1 can enable the UAC device, and keep other parameters unchanged as the corresponding ones of returned URC for other USB configuration. For example, check USB function setting first by executing **AT+QCFG="USBCFG"**, and URC **+QCFG: "USBCFG",0x2C7C,0x0125,1,1,1,1,1,0** will be returned. Then **AT+QCFG="USBCFG",0x2C7C,0x0125,1,1,1,1,1,1** should be executed to enable the UAC device on module. (This step only needs to be performed once if customers do not update the firmware.)



Note: If **AT+QCFG="USBCFG"** has only 6 function parameters, the current firmware does not support AT to enable UAC sound card device.

Step 3: Execute **AT+QPCMV=1,2** on the USB AT port to enable UAC function.

Step 4: ATD dials a call or ATA accepts the incoming call on the USB AT port.

Step 5: Audio application on the device will receive 640 bytes of PCM data through the virtual sound card every 40ms, and process (broadcast or save) the received data immediately.

Step 6: Audio application on the device will obtain microphone recording data through audio process unit. It is required that the PCM application should send 1600 bytes of voice data to the virtual sound card at an interval of 100ms.

Step 7: Execute **ATH** command to end the phone call.

Step 8: Execute **AT+QPCMV=0** to disable UAC function.

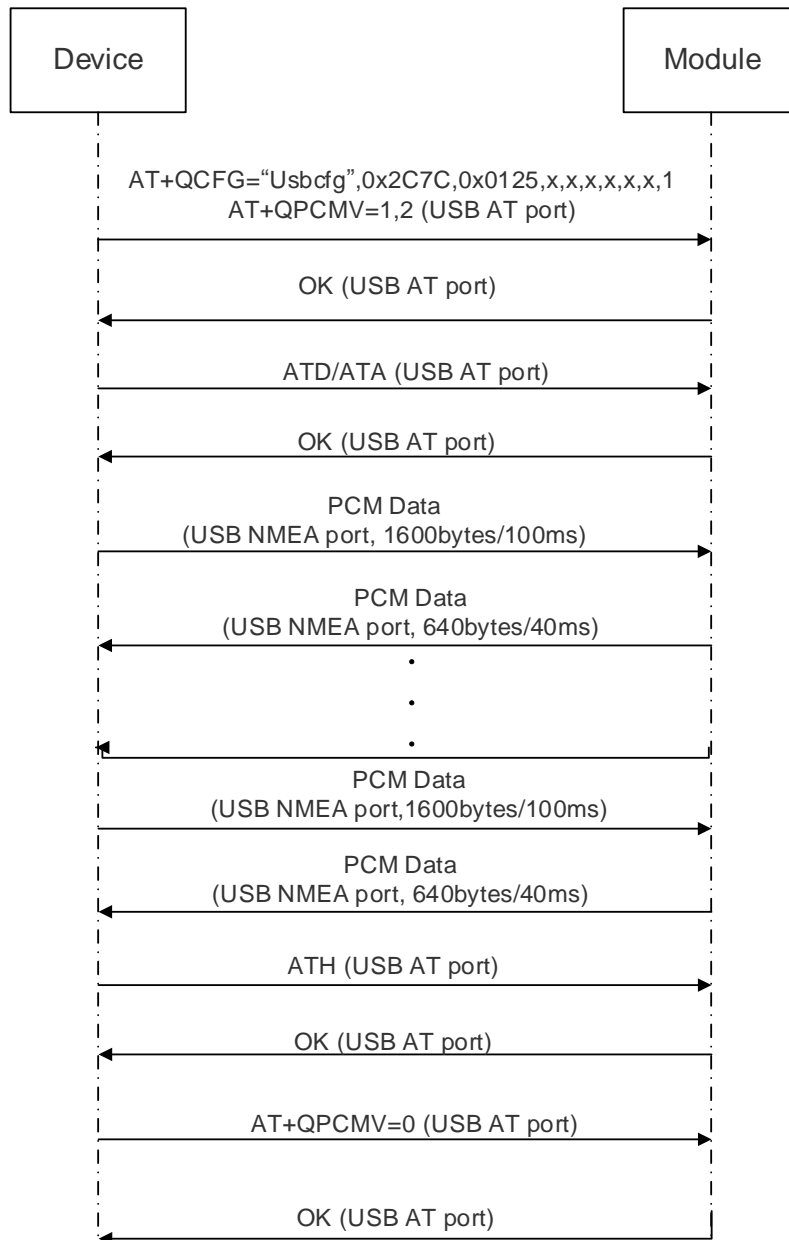


Figure 4: Operation Procedures of UAC Function

4.3. Voice over USB Use in Upper Machine

For detailed use of Voice over USB function in upper machine, please refer to **Chapter 2**.

4.4. UAC Use in Upper Machine

4.4.1. UAC Use in Windows Devices

USB sound card driver (UAC device driver) is included in Windows system. Therefore, when USB virtual sound card device is enabled on module and module USB interface is connected to Windows device, a virtual sound card device “AC Interface” will be found in the device manager. If there is no virtual sound card device but an unidentified device with a yellow exclamation point in the device manager, the USB virtual sound card driver needs to be upgraded by using the Windows built-in or a third-party driver tool.

Then customers can use the virtual sound card to play or record the voice data of the call like a normal sound card.

4.4.2. UAC Use in Linux Devices

Step 1: Add VID and PID.

In order to recognize the module, the following module VID and PID information should be added to file `[KERNEL]/drivers/usb/serial/option.c`.

```
static const struct usb_device_id option_ids[] = {  
#if 1 //Added by Quectel  
    { USB_DEVICE(0x2C7C, 0x0121) }, /* Quectel EC21 */  
    { USB_DEVICE(0x2C7C, 0x0125) }, /* Quectel EC25/EC20 R2.0/EC20 R2.1 */  
    { USB_DEVICE(0x2C7C, 0x0191) }, /* Quectel EG91 */  
    { USB_DEVICE(0x2C7C, 0x0195) }, /* Quectel EG95 */  
#endif
```

Also the following information should be added for different Linux kernel versions.

For Linux kernel version newer than 2.6.30, the following information should be added to file `[KERNEL]/drivers/usb/serial/option.c`.

```
static int option_probe(struct usb_serial *serial, const struct usb_device_id *id) {  
    struct usb_wwan_intf_private *data;  
    .....  
#if 1 //Added by Quectel  
//Quectel EC25&EC21&EC20 R2.0&EC20 R2.1&EG91&EG95's interface 4 can be used as USB
```


sound card device

```
if (serial->dev->descriptor.idVendor == cpu_to_le16(0x2C7C)
    && serial->interface->cur_altsetting->desc.bInterfaceNumber >= 4)
    return -ENODEV;
```

```
#endif
```

```
/* Store device id so we can use it during attach. */
usb_set_serial_data(serial, (void *)id);
return 0;
}
```

For Linux kernel version older than 2.6.31, the following information should be added to file `[KERNEL]/drivers/usb/serial/option.c`.

```
static int option_startup(struct usb_serial *serial)
{
.....
dbg("%s", __func__);
#if 1 //Added by Quectel
//Quectel EC25&EC21&EC20 R2.0&EC20 R2.1&EG91&EG95's interface 4 can be used as USB
sound card device
    if (serial->dev->descriptor.idVendor == cpu_to_le16(0x2C7C)
        && serial->interface->cur_altsetting->desc.bInterfaceNumber >= 4)
        return -ENODEV;
#endif
.....
}
```

For more details, please refer to **document [1]**.

Step 2: Enable USB sound function, and rebuild Linux kernel.

Run `make menuconfig` command, then enable USB sound function in Linux kernel menuconfig interface by choosing the following options in turn.

- [*] Device Drivers →
 - [*] Sound card support →
 - [*] Advanced Linux Sound Architecture →
 - [*] USB sound devices →

```
.config - Linux/arm 4.14.52 Kernel Configuration
Device Drivers  Sound card support  Advanced Linux Sound Architecture
Advanced Linux Sound Architecture
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----).
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in
[ ] excluded <M> module < > module capable

(-)
[*]  Verbose procfs contents
[ ]  Verbose printk
[ ]  Debug
<*> Sequencer support
< > Sequencer dummy client
<*> OSS Sequencer API
[*]  Use HR-timer as default sequencer timer
[*]  Generic sound devices --->
    HD-Audio ----
(64) Pre-allocated buffer size for HD-audio driver
[*]  ARM sound devices ----
[ ]  SPI sound devices ----
[*]  USB sound devices --->
<*> ALSA for SoC audio support --->

<Select>  < Exit >  < Help >  < Save >  < Load >
```

Then rebuild Linux kernel and reboot the device.

Step 3: Use and verify UAC function.

Restart customers' Linux device and insert the module USB port into it. Then new sound card device will be found in path `/dev/snd`, and customers can use it like a normal sound card device through a third-party audio tool or a third-party audio library. But the PCM data format is fixed at 8KHz sampling rate, 16bit linear and little-endian.

The source code for tinyalsa tool can be acquired from <https://github.com/tinyalsa/tinyalsa>.

Run the following two commands with tinyalsa tool to play and record the voice respectively. The "x" in the commands are the sound card number of customers' device.

Voice play:

```
tinypplay music.wav -D x -d 0 -c 1 -r 8000
```

Voice recording:

```
tinycap rec.wav -D x -d 0 -c 1 -r 8000
```

5 Appendix A References

Table 1: Related Documents

SN	Document Name	Remark
[1]	Quectel_WCDMA<E_Linux_USB_Driver_User_Guide	Linux USB Driver User Guide for UC15, UC20, EC25, EC21, EC20, EG91, EG95 and EG06 modules

Table 2: Terms and Abbreviations

Abbreviation	Description
DSP	Digital Signal Processing
NMEA	National Marine Electronics Association
PCM	Pulse Code Modulation
UAC	USB Audio Class
USB	Universal Serial Bus